A99-16873

AIAA 99-1076

Approximating Interface Topologies with Applications for Interface Tracking Algorithms

M. Williams and D. Kothe
Los Alamos National Laboratory
Los Alamos, NM

E.G. Puckett
University of California
Davis, CA
Department of Mathematics

37th AIAA Aerospace Sciences Meeting and Exhibit
January 11-14, 1999 / Reno, NV

Approximating Interfacial Topologies with Applications for Interface Tracking Algorithms *

M. W. Williams † D. B. Kothe ‡
Los Alamos National Laboratory
Los Alamos, New Mexico, 87545

E. G. Puckett §
University of California, Davis
Davis, California, 95616

Abstract

We incorporate smooth kernels and convolution theory into continuum surface force models. We convolve a discontinuous scalar function with the first order spatial derivatives of the kernel to determine the unit normal to an interface. This algorithm is applicable to unstructured three dimensional grids. We compare these results to a "least squares" approach for several simple geometries and demonstrate that these normals in conjunction with a piecewise planar interface reconstruction algorithm rapidly approaches linear preservation. We discuss the interface reconstruction algorithm and a method for determining the area of the interface plane in each cell. We also demonstrate an effective velocity filter for removing some of the nonphysical modes in the velocity field which can be induced by numerical methods.

Introduction

We begin with a brief discussion of the nomenclature used herein. The examples used in this report use two immiscible fluids. The scalar function, f, denotes the characteristic function of one of these fluids. This function can be represented by the volume fraction distribution of this fluid and is commonly termed the "color function". The function f can be smoothed (or mollified) several ways, (see [WKP99]), the smoothed color function will denoted by $\tilde{f}$. In this report we mollify f by convolving the scalar function with a kernel, denoted $\boldsymbol{K}$. In three dimension the kernel is nonzero with a spherical region known as the *support* of the kernel. The radius of this sphere is known as the *radius of support* and is denoted by ϵ. The unit normal to an interface is denoted $\hat{\mathbf{n}}$.

Approximating Unit Normals

One the most crucial steps to efficient and accurate interface tracking algorithms is the accurate approximation to the unit normals of the interface. We review a method for determining the unit normals presented in [WKP99] and used in [AP95]. This method will be referred to as the "convolution method". We incorporate this method into a continuum surface force (CSF) method, originally presented in [BKZ92]. We then show results for unit normals calculated on some typical geometries.

Given a mollified color function, $\tilde{f}$, the unit normals are found by

* Supported by the U.S. DOE Accelerated Strategic Computing Initiative (ASCI) program.
A updated version of this report can be found at http://lune.mst.lanl.gov/Telluride/Text/publications.html
†Structure/Property Relations Group MST–8, MS-G755, *mww@lanl.gov.*
‡Structure/Property Relations Group MST–8, MS-G755, *dbk@lanl.gov.*
§Department of Mathematics, *egp@math.ucdavis.edu.*

$$\hat{\mathbf{n}} = \frac{\nabla \tilde{f}}{|\nabla f|}. \quad (1)$$

Each of the components of the gradient is found by convolving the scalar function f with the first derivative of some kernel, $\boldsymbol{K}$, e.g.

$$n_x = \frac{\partial \tilde{f}(x)}{\partial x} = \int_\Omega \frac{\partial \boldsymbol{K}}{\partial x}(\boldsymbol{x}'-\boldsymbol{x}) f(\boldsymbol{x}') d\boldsymbol{x}'. \quad (2)$$

Similar equations determine n_y and n_z.

Equation (2) is found numerically by using a simple midpoint rule to approximate the integral. With minor effort, equation (2) can also be approximated on unstructured grids. This method was applied to determine the unit normals to an inclined plane and a spherical drop using both structured and unstructured meshes. The results are presented in Tables 1, 3 and Tables 4, 6.

The results in Tables 1 and 4 show that the convolution method produces unit normal approximations which are better than first order accurate, though not quite second order in the L^∞ norm. For unstructured grids (Tables 3 and 6, the method produces extremely nice results; doubling the number of mesh cells can reduce the L^∞ errors by a factor of 4. Keep in mind that these are three dimensional problems and for a uniform grid doubling the mesh size is equivalent to reducing the cell size by about 80%. A crude estimation for the unstructured grid implies the algorithm is well above second order accurate. The results using a "generalized" least squares linear regression (LSLR) method are also give for reference. By "generalized" here we mean that the LSLR method is also applicable for unstructured grids. Note that the results for this method offer a lower error for the coarse grid calculations yet it fails to converge in the L^∞ norm.

Δx	ϵ	L^1 Error	L^∞ Error
0.100	0.150	5.6820E-03	0.19010
0.050	0.150	0.1310E-03	0.00828
0.025	0.120	0.0025E-03	0.00300

Table 1: Errors in calculated unit normals for a plane inclined at 30 degrees, using the convolution method.

Δx	L^1 Error	L^∞ Error
0.100	2.0597E-03	0.5044E-01
0.050	1.5474E-03	0.5109E-01
0.025	0.6593E-03	0.5109E-01

Table 2: Errors in calculated unit normals for a plane inclined at 30 degrees, using a least squares linear regression algorithm.

# of cells	L^1 Error	L^∞ Error
7454	4.0766E-03	0.3317
14298	1.2738E-03	0.1467
33153	0.3610E-03	0.0468

Table 3: Errors in calculated unit normals for a plane inclined at 30 degrees, using the convolution method on an unstructured tetrahedron mesh with $\epsilon = 0.15$.

Δx	ϵ	L^1 Error	L^∞ Error
0.100	0.15	12.984E-03	3.8847E-01
0.050	0.15	0.2699E-03	0.1420E-01
0.025	0.10	0.0416E-03	0.0550E-01

Table 4: Errors in calculated unit normals for a spherical drop, $R = 0.25$, using the convolution method.

Δx	L^1 Error	L^∞ Error
0.100	2.1898E-03	0.3434E-01
0.050	1.1169E-03	0.6894E-01
0.025	0.6609E-03	0.7958E-01

Table 5: Errors in calculated unit normals for a spherical drop, $R = 0.25$, using a least squares linear regression algorithm.

# of cells	L^1 Error	L^∞ Error
7454	5.8950E-03	0.2193
14298	1.3982E-03	0.1038
33153	0.3890E-03	0.0266

Table 6: Errors in calculated normals for a spherical drop, $R = 0.25$, on an unstructured tetrahedron mesh with $\epsilon = 0.15$.

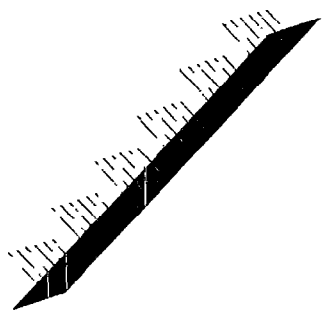


Figure 1: Side view of an interface reconstruction and normal approximations to a plane on a 10^3 mesh. The plane is tilted at 45 degrees, normals were found at cell centers and the algorithm preserves linearity for all uniform grids. For this example $\epsilon = 0.150$.

Preserving Linearity

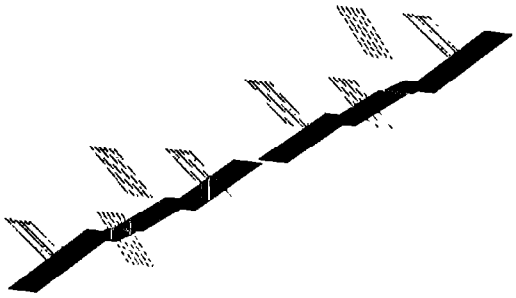
The term "linearity preserving" refers to an interface reconstruction algorithm which can exactly reproduce a linear interface, i.e. a line for two dimensions or a planar interface for three dimensions. The interface reconstruction algorithm used here is a piecewise planar reconstruction which reproduces a planar approximation to the interface in each cell containing the interface. The unit normal, $\hat{\mathbf{n}}$, determines the orientation of the plane while the placement of the plane is determined by the volume fraction, f of the cell. The interface reconstruction algorithm takes $\hat{\mathbf{n}}$ and returns a value of ρ for the equation of a plane given by

$$\hat{\mathbf{n}} \cdot x = \rho \tag{3}$$

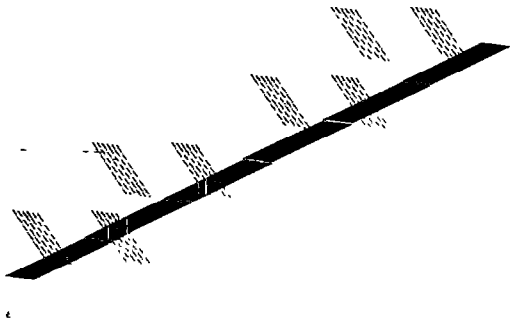
which satisfies the volume fraction for each cell containing the interface.

The algorithm for determining normals discussed in the previous section can reproduce linear interfaces exactly but only for special circumstances. If the true planar interface passes through cell centers then this method is indeed linearity preserving when used with a radially symmetric kernel. An example of this feature is plotted in Figure 1.

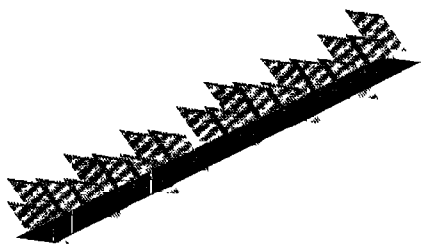
While a quantitative estimate of linear preservation is difficult to obtain, figures 2 and 3 imply that this algorithm converges rapidly to one which preserves linearity on a uniform mesh. The thin white lines across the plane in these figures demonstrate a lack of linear preservation. Note that a larger radius of support for the kernel, $\boldsymbol{K}$, has a dramatic effect on linear preservation for coarse grids (see Figures 2(a-b) and 3(a-b)). Figure 4 examines linear preservation on an unstructured mesh. Just as one might expect, the figures shows that linear preservation may be more difficult to obtain when using this scheme on unstructured grids.



(a) A 10^3 mesh with $\epsilon = 0.150$.



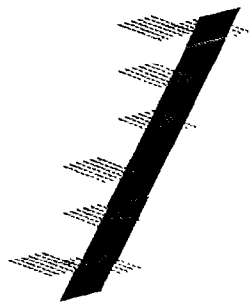
(b) A 10^3 mesh with $\epsilon = 0250$.



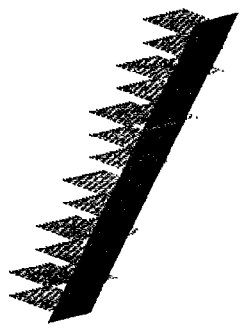
(c) A 20^3 mesh with $\epsilon = 0.150$.

Figure 2: The interface reconstruction and normal approximations to a plane in the unit cube. The plane is tilted at 30 degrees, normals were found at cell centers and the algorithm rapidly approaches linear preservation.

(a) A 10^3 mesh with $\epsilon = 0.150$.

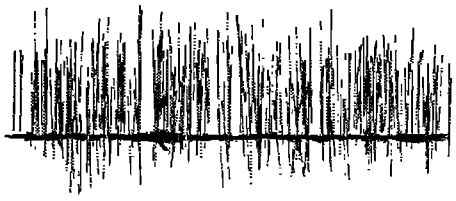


(b) A 10^3 mesh with $\epsilon = 0250$.

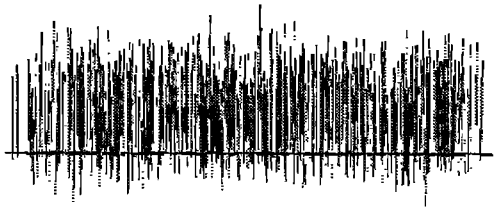


(c) A 20^3 mesh with $\epsilon = 0.150$.

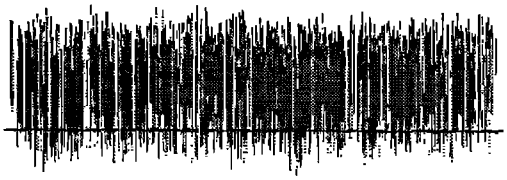
Figure 3: The interface reconstruction and normal approximations to a plane in the unit cube. The plane is tilted at 67 degrees, normals were found at cell centers and the algorithm rapidly approaches linear preservation.



(a) 7454 cell tetrahedron mesh.



(b) 14298 cell tetrahedron mesh.



(c) 33153 cell tetrahedron mesh.

Figure 4: Side view of an interface reconstruction and normal approximations to a plane on 3D unstructured meshes. The plane was originally tilted at 45 degrees, $\epsilon = 0.15$ was used for all meshes. Normals were found at cell centers and the algorithm does not preserve linearity.

Reconstructing Interfaces

The interface reconstruction used for this report is a piecewise planar reconstruction. There are various reasons why one may want to reconstruct the geometry of the interface within a cell, e.g. visualization or to apply forces acting on the interface such as surface tension forces or drag due to viscous fluid motion. For each cell, the plane given by the values of $\hat{\mathbf{n}}$ and ρ determines which edges of the cell are intersected by the planar interface approximation. These points of intersection are the vertices of the polygon which represents the planar interface for that cell. An example of a polygon outlined by the planar interface reconstruction within an unstructured cell is shown in Figure 5. Once an interior reference point (P_1) is established, the polygon can be reconstructed as conjoined triangles where the polygon vertices and P_1 form the vertices for each triangle.

It may also be necessary to order the vertices in some fashion, e.g. for a reconstruction for interface visualization. This can be accomplished by determining the magnitude of the angle between some reference vector (V_1) and the vector formed by the vertex and the reference point (see Figure 5); the vertices are then ordered by the magnitude of this angle, Θ_n. The area of the interface is easily found as the summed area of all of the triangles. The reconstructed interface can be quite discontinuous. Figure 6 shows a typical planar interface reconstruction in two adjacent unstructured cells. For a given volume fraction distribution, one can see that the interface in surrounding cells has no effect on the final geometry of the plane reconstructed in some given cell.

Figures 7 and 8 show examples of an interface reconstruction for some simple geometries on unstructured meshes. The figures also show the outlines of the cells which

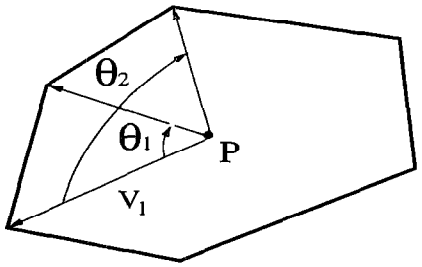


Figure 5: Example of a typical interface reconstruction within a cell. Planar interfaces are reconstructed as conjoined triangles.

Δx	Calc. Area	% Error
0.100	0.7863	0.1148
0.050	0.7844	0.1271
0.025	0.7851	0.0380

Table 7: Calculated surface areas and errors for a spherical drop, $R = 0.25$, using $\epsilon = 0.15$.

contain the interface. As expected, the figures become sharper and more defined as the number of cells is increased. The calculated areas for these geometries is shown in Tables 7 - 9. Tables 7 and 8 show results for the calculation of the surface area of a spherical drop for structured and unstructured meshes. The estimation of areas is nearly second order for uniform meshes, while the results for the unstructured mesh are inconclusive. The calculated surface areas for a cube on an unstructured mesh are shown in Table 9. These results show the results to be well over second order accurate. It should be noted that the results for the cube will not converge to the exact solution for a constant ϵ, since the convolution method will tend to "smooth" the sharp corners.

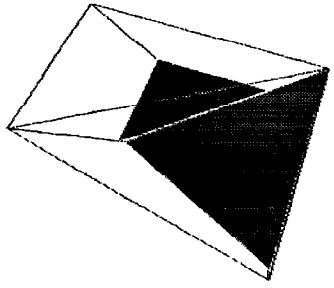


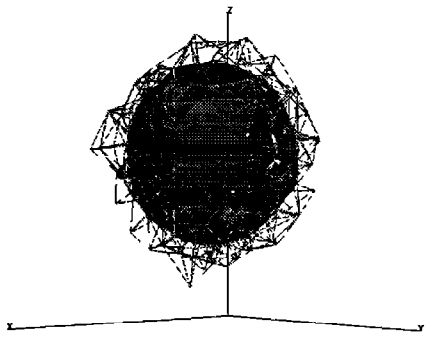
Figure 6: Example typical of a discontinuous planar interface reconstruction in two adjacent unstructured cells.

# of cells	Calc. Area	% Error
7454	0.8278	5.3988
14298	0.7851	0.0380
33153	0.7867	0.1656

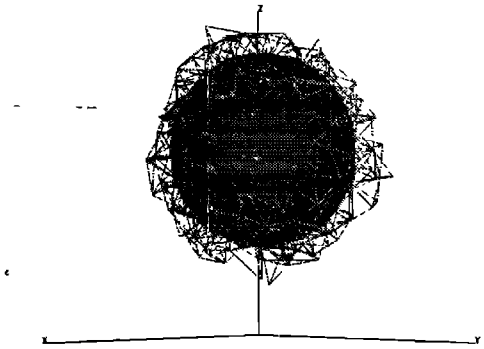
Table 8: Calculated surface areas and errors for a spherical drop, $R = 0.25$, on an unstructured tetrahedron mesh, using $\epsilon = 0.15$.

# of cells	Calc. Area	% Error
7454	0.9697	1.010
14298	0.9542	0.604
33153	0.9595	0.052

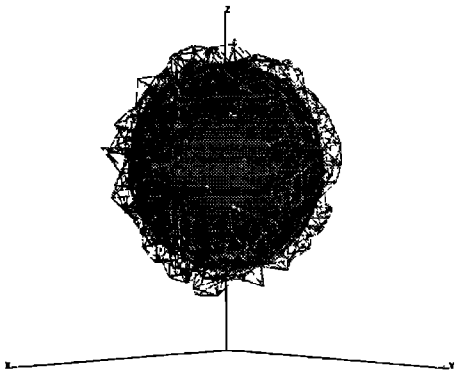
Table 9: Calculated surface areas and errors for a cube, $0.4X0.4X0.4$, on an unstructured tetrahedron mesh, using $\epsilon = 0.15$.



(a) 7454 cell tetrahedron mesh.

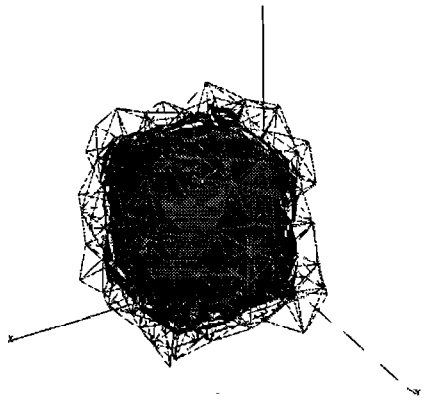


(b) 14298 cell tetrahedron mesh.

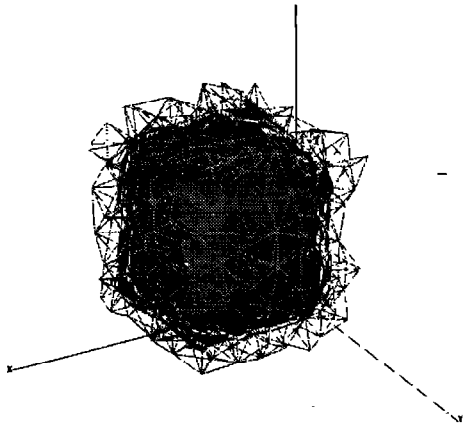


(c) 33153 cell tetrahedron mesh.

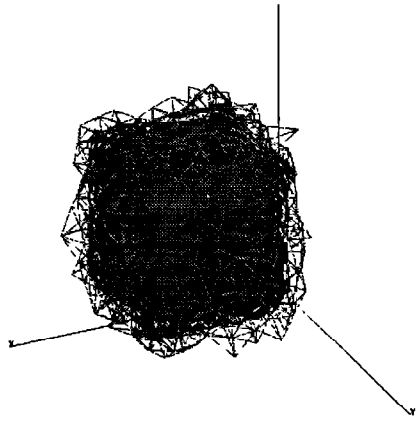
Figure 7: Planar interface reconstruction of a sphere on an unstructured mesh. The outlines of cells containing the interface are also shown.



(a) 7454 cell tetrahedron mesh.



(b) 14298 cell tetrahedron mesh.



(c) 33153 cell tetrahedron mesh.

Figure 8: Planar interface reconstruction of a cube on an unstructured mesh. The outlines of cells containing the interface are also shown.

Velocity Filter

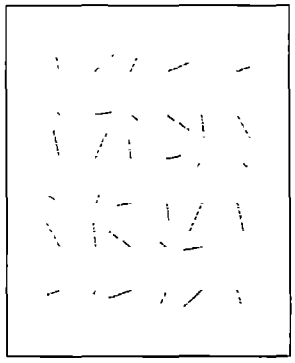
Numerical models inevitably introduce nonphysical modes into the velocity field when modeling fluid flows with large density ratios (see [Rid98]). This is one reason why it is important to implement some form of "filter" to dampen these high frequency nonsolenoidal modes. Here we have implemented a vertex-based velocity filter which detects several nonsolenoidal modes of the velocity field not "seen" by the cell centered algorithm. The filter subtracts a correction term from the current velocity field, $\tilde{\mathbf{u}}$. A correction term is found by solving 4, where $\nabla \cdot \tilde{\mathbf{u}}_v$ is the divergence of the velocity found at the cell vertices.

$$\phi = (\Delta)^{-1} \nabla \cdot \tilde{\mathbf{u}}_v \tag{4}$$

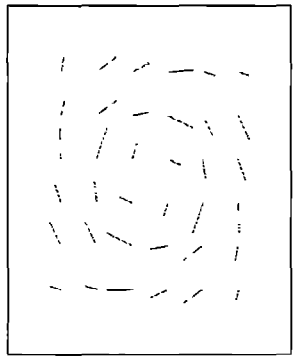
The correction term is then applied to yield a new velocity, $\mathbf{u}$, here $\frac{1}{\rho}\nabla \cdot \phi_c$ is the cell centered correction term.

$$\mathbf{u} = \tilde{\mathbf{u}} - \frac{1}{\rho}\nabla \cdot \phi_c \tag{5}$$

Figure 9 shows a vortex velocity field with nonsolenoidal noise added to the velocity after one time step with and without the implementation of the velocity filter. Figure 9(b) shows that the filter is quite effective in eliminating the artificially induced, nonphysical components from the velocity field.



(a) Without filter



(b) With filter

Figure 9: Vortex velocity field with nonsolenoidal "noise" added. Results show one time step evolution with and without filtering.

Conclusions

We have demonstrated that the convolution method can produce better than second order accurate normals to an interface for some given geometries. This method, in conjunction with a piecewise planar interface reconstruction algorithm, gives results which demonstrate a strong convergence towards a linear preserving algorithm. These components (normal approximation and interface reconstruction) may also be implemented on unstructured meshes without severe detrimental effects. Although the interface reconstruction can be quite discontinuous, the method shows convergence to well defined interfaces upon mesh refinement. The interface area calculations show themselves to be nearly second order for some cases. The vertex based velocity filter has been shown to be effective for removing nonsolenoidal modes in the velocity field.

References

[AP95] I. Aleinov and E. G. Puckett. Computing surface tension with high-order kernels. In H. A. Dwyer, editor, *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*, pages 13–18, Lake Tahoe, NV, 1995.

[BKZ92] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.

[KRM96] D. B. Kothe, W. J. Rider, S. J. Mosso, J. S. Brock, and J. I. Hochstein. Volume tracking of interfaces having surface tension in two and three dimensions. Technical Report AIAA 96–0859, AIAA, 1996. Presented at the 34rd Aerospace Sciences Meeting and Exhibit.

[Rid98] W. J. Rider. Filtering nonsolenoidal modes in numerical solutions of incompressible flows. Technical Report LA–UR–94–3014, Los Alamos National Laboratory, 1998. Accepted for publication in the *International Journal for Numerical Methods in Fluids.*

[RKP98] W. J. Rider, D. B. Kothe, E. G. Puckett, and I. D. Aleinov. Accurate and robust methods for variable density incompressible flows with discontinuities. In V. Venkatakrishnan, M. D. Salas, and S. R. Chakravarthy, editors, *Workshop on Barriers and Challenges in Computational Fluid Dynamics*, pages 213–230, Boston, MA, 1998.[1] Kluwer Academic Publishers.

[WKP99] M.W. Williams, D. B. Kothe, and E. G. Puckett. Convergence and accuracy of kernel-based continuum surface tension models. In W. Shyy, H. Udaykumar, M. Rao, and R. Smith, editors, *Computational Fluid Dynamics with Moving Boundaries.* Cambridge Publishing, 1999.